The WHUTE System in NTCIR-9 RITE Task

Han Ren
School of Computer, Wuhan University, Wuhan 430072, China
hanren@whu.edu.cn

Chen Lv
School of Computer, Wuhan University, Wuhan 430072, China
lvchen1989@gmail.com

Donghong Ji
School of Computer, Wuhan University, Wuhan 430072, China
dhji@whu.edu.cn

ABSTRACT
This paper describes our system of recognizing textual entailment for RITE Chinese subtask at NTCIR-9. We build a textual entailment recognition framework and implement a system that employs string, syntactic, semantic and some specific features for the recognition. To improve the system's performance, a two-stage recognition strategy is utilized, which first judge entailment or no entailment, and then contradiction or independence of the pairs in turn. Official results show that our system achieves a 73.71% performance in BC subtask, 60.93% in MC subtask and 48.76% in RITE4QA subtask.

Keywords
Recognizing Textual Entailment, Binary-Class Subtask, Multi-Class Subtask, Two-Stage Recognition

1. INTRODUCTION
Given a text fragment, the goal of Recognizing Textual Entailment(RTE) is to recognize a hypothesis that can be inferred from it or not. RTE is a notable field of research that is leveraged in many natural language processing areas, i.e., document summarization, question answering and machine translation[6, 8, 9].

This year, NTCIR-9 evaluation workshop defines a new RTE task named RITE, which evaluates systems that automatically detect entailment, paraphrase and contradiction in texts written in Japanese, Simplified Chinese or Traditional Chinese. The first RITE task defines four subtasks, Binary-Class(BC), Multi-Class(MC), Entrance Exam and RITE4QA subtask. We participate in three subtasks, BC, MC and RITE4QA subtask of Simplified Chinese out of them, and submit three results(RITE1-WHUTE-CS-BC-01, RITE1-WHUTE-CS-BC-02 and RITE1-WHUTE-CS-BC-03) for BC subtask, two results(RITE1-WHUTE-CS-MC-01 and RITE1-WHUTE-CS-MC-02) for MC subtask and three results(RITE1-WHUTE-CS-RITE4QA-01, RITE1-WHUTE-CS-RITE4QA-02 and RITE1-WHUTE-CS-RITE4QA-03) for RITE4QA subtask.

Since the task definition of RITE is similar with that of RTE series challenges, the system we implemented in the RTE-5 challenge can be easily modified for RITE task. In our previous system[10], the classifier is trained to assign each pair to one of three types(Entailment, Contradiction and Unknown). In our system of the RITE task, the types are extended to five(Forward, Reverse, Bidirection, Contradiction and Independence), and some directed features are duplicated, in MC subtask, to compute the bidirectional similarity of text(T) and hypothesis(H) in each pair respectively. For a better performance, we also utilize a two-stage recognition strategy, which first judge entailment or no entailment, and then contradiction or independence from the pairs in turn.

Our system employs Support Vector Machine(SVM) for classification. We also make use of string similarity measures as well as syntactic and semantic similarity measures to build the features for training and prediction. On the other hand, although large resources and background knowledge bases such as paraphrase collections and geographic ontologies contribute to a better performance[3, 12], the available resources of Chinese for recognizing textual entailment are still lacking so that we only employ some basic resources, i.e., PropBank and Named Entity Recognizer, for the linguistic-based features.

The rest of this paper is organized as follows. In section 2, the architecture and workflow of the system are described. Section 3 gives a more detailed explanation for each part of the system, including preprocessing, the features and the two-stage recognition strategy. Section 4 discusses the experimental results and error analysis. Finally, some conclusions and the future work are given.

2. SYSTEM OVERVIEW
The overall architecture of system is shown in Figure 1, which contains a preprocessing model, a feature extraction model and two classifiers. In the feature extraction model, the training and testing models utilize the same features. Procedures of the system is described as follows:

1) For each text fragment and hypothesis, a preprocessing procedure is performed, including word segmentation, part-of-speech tagging, named entity recognition, syntactic dependency parsing and semantic role labeling.

2) In feature extraction, string features, syntactic features and semantic features are computed. Named entity relations are also considered.

3) All features are employed to classify entailment against no entailment, and then contradiction against independence for each pair.

3. DESCRIPTION OF THE SYSTEM
The input data for the system is pairs of sentences, which include a text and a hypothesis for each pair, and the output is a boolean value for each pair, which indicates Entailment if the system decides that the text entails the hypothesis, or No Entailment otherwise. The system also gives the entailment confidence indicates the degree of entailment for each pair.

3.1 Preprocessing
The preprocessing procedure includes word segmentation, Part-Of-Speech(POS) tagging, named entity recognition, syntactic parsing and shallow semantic paring.

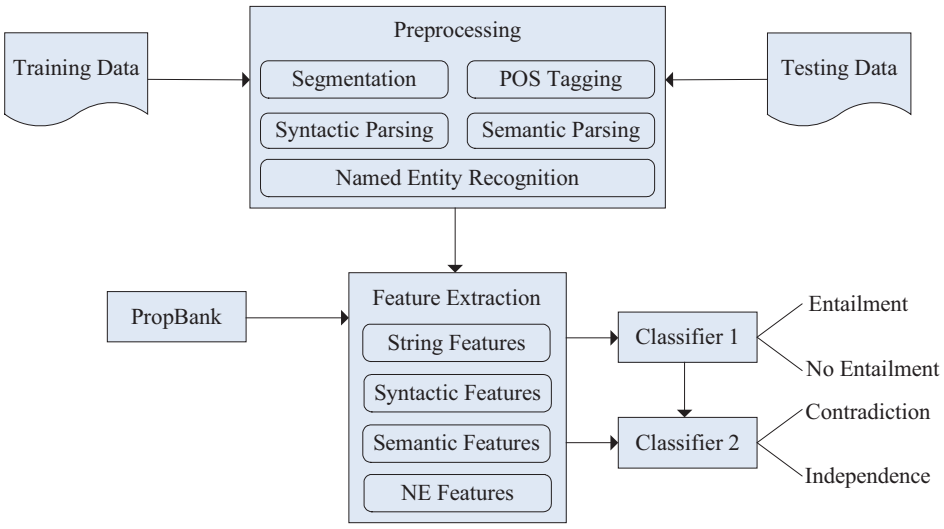

Figure 1. System architecture

Initially, the text and the hypothesis for each pair are segmented by Stanford Chinese Word Segmenter [1], a free tool employs Conditional Random Field(CRF) model as the classifier. In the segmenter, we utilize the first model follows the Chinese Penn Treebank standard, since the training data of syntactic and shallow semantic parsing that comes from CoNLL2009 Shared Task also follow this standard. The segmented results are then tagged POS by Stanford POS Tagger [2], a log-linear tagger implemented using Maximum Entropy model also follows the Chinese Penn Treebank standard. Both the tools are implemented by Java so that they are easily invoked by our system. For named entities, we only consider personal names, locations, organizations and temporal expressions, and utilize ICTCLAS[3], a free Chinese POS tagger and NE recognizer, for the recognition. However, in the experiments, we find that inconsistent numerical expressions, including some abnormal temporal expressions mixed with Chinese and Arabic numerals, give the performance a great impact. To this end, we implement a numeral normalization tool, transforming the temporal and Chinese numeral expressions to the Arabic numerals.

The syntactic and semantic parsing model follows our system in CoNLL2009[11], which labels syntactic and semantic dependency relations of words, since shallow syntactic and semantic relations are more flexible and precise. The annotation standard is identical with the definition in CoNLL2009, with 30 tags for the syntactic dependents and 25 tags for the semantic roles.

3.2 String Features

The idea of the string features is simple: if a part of *T*'s surface string is very similar to *H*'s, it is an indication that *T* may entail *H*[2]. In the series RTE challenges, rich string features are leveraged by the classification-based systems. Since one goal of our participation for RITE is meant to investigate string features that are useful for recognizing Chinese textual entailment, we select 16 features employed by most systems[1, 3, 5, 10, 12] in series RTE Challenges. The following describes the string features leveraged in our system.

N-gram Overlap The motivation of this feature is simple, considering how similar the hypothesis is to the text by comparing how many of the same n-grams appear in *H* of each pair. In our system, bigram and trigram are taken into account. The feature is computed as below:

$$Ngram\ Matching = \frac{ngram(T) \cap ngram(H)}{ngram(H)} \quad (1)$$

Word Overlap This feature is similar with the N-gram Overlap, except that bigram in the latter feature is replaced by word. Recall that the text snippets in each pair are segmented to words in the preprocessing.

Matching Coefficient Different with Word Overlap, this feature considers $|words(T) \cap words(H)|$, namely how many of the same words appear in both *T* and *H*, where $words(T)$ and $words(H)$ are the word sets of the text and the hypothesis in each pair.

Length Ratio This feature considers the length ratio of the text snippets in each pair. The length is the total number of the unigrams in each text snippet.

Jaccard Coefficient This feature considers how many of the same words appear in both *T* and *H* in each pair. The feature is computed as below:

$$Jaccard\ Coefficient = \frac{words(T) \cap words(H)}{words(T) \cup words(H)} \quad (2)$$

Dice Coefficient Dice coefficient is well known, considers how similar of *T* and *H* of each pair in our system by computing as follows:

$$Dice\ Coefficient = \frac{2 \cdot words(T) \cap words(H)}{words(T) + words(H)} \quad (3)$$

LCS Similarity This feature in our system estimates the similarity between the longest common substring of *T* and *H* in each pair, and the shorter one in two of them. It is computed as below:

[1] http://nlp.stanford.edu/software/segmenter.shtml
[2] http://nlp.stanford.edu/software/tagger.shtml
[3] http://ictclas.org/

$$LCS\ Similarity = \frac{LCS(T,H)}{min\{words(T), words(H)\}} \quad (4)$$

Cosine Similarity This feature builds the word vectors of *T* and *H* in each pair, and computes its cosine similarity.

Levenshtein Distance Also known as edit distance, this distance considers the minimum number of transform operations from one string to another, where an operation refers to an insertion, deletion or substitution of a single unit, which in our system is a Chinese character or a word.

Euclidean Distance This feature is defined as follows, where x_i and y_i correspond to t_1 and t_2 respectively:

$$Euclidean\ Distance = \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2} \quad (5)$$

Manhattan Distance This feature computes City Block distance with the following formula, where x_i and y_i correspond to t_1 and t_2 respectively:

$$Manhattan\ Distance = \sum_{i=1}^{n}|x_i - y_i| \quad (6)$$

Chebyshev Distance This feature defines the distance of two strings the greatest of their differences along any coordinate dimension.

Jaro Distance This metric proposed by Jaro[7] mainly estimates duplicate degree between two strings.

Jaro-Winkler Distance This metric proposed by Winkler[13] modifies Jaro Distance by assigning a higher weight to the two strings that are more similar with their prefixs.

Minimal Substring Similarity This metric considers the minimal Jaro-Winkler Distance between the substring of text and hypothesis in each pair.

Maximal Substring Similarity Similar with Minimal Substring Similarity, but computes the max Jaro-Winkler distance between the substring of the text and the hypothesis.

3.3 Syntactic Features

Three syntactic features are employed in our system, aiming at estimating similarity of the dependency structures between the text and the hypothesis in each pair.

Unlabeled Sub Tree Overlap This features computes the ratio of the same sub trees in the text and the hypothesis, as described in the following formula. Each sub tree has a head and one of its dependents derived from the dependency tree. Figure 2 shows an example of the sub trees in a sentence. Two sub trees are viewed identical if they have the same heads and the corresponding dependents.

$$UST\ Overlap = \frac{subtree(T) \cap subtree(H)}{subtree(H)} \quad (7)$$

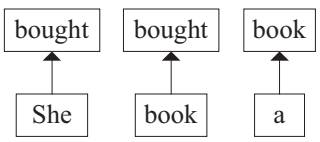

Figure 2. Sub trees in the sentence "She bought a book". The arrow in each sub tree denotes the dependency direction.

Labeled Sub Tree Overlap Similar with Unlabeled Sub Tree Overlap, this feature also consider how similar the hypothesis is to the text by comparing the ratio of the same sub trees appear in *H*, except that the dependency relations(or classes) are also taken into account in sub trees.

Partial Sub Tree Overlap In comparison with the above features, this feature is more relax, taking partial matching of the sub trees into account. That is, two sub trees are viewed partially identical if they have the same heads or the dependents. In order to differ full matching and partial matching of sub trees, we set a weighting value, which equals 1 if sub trees are full matched, 0.5 if partially matched and 0 if no matched.

3.4 Semantic Features

Only one shallow semantic feature is employed in our system, aiming at estimating similarity of the semantic structures between the text and the hypothesis in each pair.

Predicate Argument Overlap This features computes the ratio of the same predicate-argument pairs in the text and the hypothesis. Each predicate-argument pair has a predicate and one of its arguments(if have) derived from the semantic parsing result. Two predicate-argument pairs are viewed identical if they have the same heads and the corresponding dependents. The feature is computed as below:

$$PA\ Overlap = \frac{pred\text{-}arg(T) \cap pred\text{-}arg(H)}{pred\text{-}arg(H)} \quad (8)$$

3.5 Specific Features

Since some indicators in sentences such as named entities and numerals contribute to a better performance for recognizing textual entailment[4], we also employ some specific features in our system.

Named Entity Coverage This feature gives a boolean value, where it is true if all the named entities in the hypothesis(if have) also appear in the text, or false otherwise.

Numeral Coverage This feature gives a boolean value, where it is true if all the numerals in the hypothesis(if have) also appear in the text, or false otherwise.

3.6 Two-stage Recognition Approach

In the MC subtask, not only entailment direction, namely t_1 entails t_2 or is entailed by t_2, but also entailment class, i.e., entailment or contradiction, needs to be judged. To this end, directed features such as word overlap and sub tree overlap feature are duplicated, considering t_1 is the text and t_2 is the hypothesis, and then t_2 the hypothesis and t_1 the text. The intuitive reason for this is that, if a feature gets a high score under the condition that t_1 is the text and t_2 is the hypothesis, whereas the feature gets a low one under the condition that t_2 is the text and t_1 is the hypothesis, it probably indicates that t_1 entails t_2 and not vice versa.

The system implemented employs all directed and undirected features mentioned in section 3.2-3.5. However, the performance of the experiment is no satisfying, and the main reason is obvious, since the classifier judges not only entailment class but also entailment direction at the same time. As a matter of fact, every directed feature and its duplicated one for the bidirectional judgment give an impact on the performance of classification,

hence the performance of a bidirectional recognition approach is worse than a unidirectional one, and this conclusion can be drawn according to the following experiments.

In our system, a two-stage unidirectional recognition strategy is utilized, that is, a text pair is first judged entailment or no entailment, and then contradiction or independence. More specifically, each pair $(t_1,\ t_2)$ is first transformed into two pairs $(t_1,\ t_2)$ and $(t_2,\ t_1)$, then a bi-categorization classifier is employed to judge each pair whether the left text fragment entails the right one. Thus the problem is equivalent with that of the 2-way task in RTE-5 challenge. After that, a logical decision is made, where $(t_1,\ t_2)$ has a forward entailment relation if t_1 entails t_2 but t_2 not entails t_1, or it has a reverse entailment relation on the reverse situation, or it has a bidirection relation when t_1 and t_2 are entailed to each other. If none of any entailment relation exists between t_1 and t_2, the pair $(t_1,\ t_2)$ is thrown into another bi-categorization classifier to judge if t_1 and t_2 have a contradiction or independence relation.

In the two-stage recognition approach, two classifiers for two-stage entailment judgment is trained, and the features employed in the prior one-stage system are still utilized except for those duplicated ones. For the purpose of the unidirectional entailment recognition, each pair in training data, which at least has one entailment relation from one text to another, are split into two entailment pairs. For example, if t_1 and t_2 in a pair have the relation of bidirection, then two entailment pairs $t_1 \rightarrow t_2$ and $t_2 \rightarrow t_1$ are generated automatically; if they have the relation of reverse, then $t_1 \nrightarrow t_2$ and $t_2 \rightarrow t_1$ are generated. For contradiction and independence relation, only one entailment pair is generated, that is, if t_1 and t_2 have the relation of contradiction or independence, then $t_1 \xrightarrow{C} t_2$ or $t_1 \xrightarrow{I} t_2$ is generated. Actually, this benefits the improvement of system performance, since the training data are expanded, although unessentially.

4. Experimental Results and Analysis

In RITE subtask, the number of five categories are 92 for forward, 88 for bidirection, 85 for reverse, 72 for contradiction and 70 for independence. We participate in three subtasks, BC, MC and RITE4QA subtask of Simplified Chinese.

4.1 BC Subtask

For BC subtask, we submit three runs: RITE1-WHUTE-CS-BC-01, RITE1-WHUTE-CS-BC-02 and RITE1-WHUTE-CS-BC-03. The first run utilizes only string and specific features, the second run adds syntactic features, and the third run employs all features, including semantic features. Table 1 shows the official results of the three runs.

Table 1. Official results for BC subtask

	Run-1		Run-2		Run-3	
	P	R	P	R	P	R
E	0.764	0.802	0.777	0.821	0.779	0.829
N	0.603	0.549	0.636	0.569	0.646	0.569

In Table 1, E represents entailment and N no entailment, while P represents precision and R recall. Apparently, Run-3 achieves a best performance, no matter precision or recall, as shown in Table 1. On the other hand, for entailment relation, run-2 increases a 1.3% performance of precision and a 1.9% performance of recall than run-1, while run-3 increases a corresponding 0.2% and a 0.8% than run-2; for no entailment relation, run-2 increase a 3.3% performance of precision and a 2.0% performance of recall than run-1, while run-3 only increase a corresponding 1.0% performance of precision than run-2. It indicates that the syntactic features improve the performance of our system, especially for no entailment judgment. The semantic features also improve our system's performance, but their contribution is limited, less than the syntactic features, and unhelpful for no entailment relation. As a matter of fact, two text fragments that are similar with semantic structures are often similar with syntactic ones, while if two text fragments that have the same meaning are not similar with syntactic structures, they probably are not similar with semantic ones.

Errors that occur in BC experiments lie in two folds: 1) pairs that have a high similarity between the text and the hypothesis in them are incorrectly judged. Take the pair 104 as an example, the text is '华特迪士尼得到奥斯卡特别成就奖' 'Walt Disney received Oscar Special Achievement Award', and the hypothesis is '华特迪士尼获颁金球奖特别成就奖' 'Walt Disney was awarded Golden Globe Special Achievement Award'. The two strings are almost the same with each other whatever in string or in syntactic and semantic structures, despite that the Named Entity Coverage feature can indicate the difference between them. In order to judge these pairs correctly, weights of the syntactic, semantic and some specific features such Named Entity Coverage should be improved. 2) pairs that have a low similarity are also judged incorrectly. Like the pair 82 and 83, the same words are few between the text and the hypothesis in each pair, so that the judgment for them is no entailment. In order to improve the performance, some lexical resources such as synonym and abbreviation lexicons should be employed to recognize the words having the same meaning. Meanwhile, lexical or syntactic alignment also can be utilized to improve the performance of the entailment recognition of deep semantic relation.

4.2 MC Subtask

For MC subtask, we submit two runs: RITE1-WHUTE-CS-MC-01 and RITE1-WHUTE-CS-MC-02. The first run utilizes the one-stage recognition approach, namely judges the entailment class directly by using a single classifier. The second run utilizes the two-stage recognition strategy, where two classifiers are trained for two stage recognition. Table 2 shows the official results of the two runs.

Table 2. Official results for MC subtask

	Run-1		Run-2	
	P	R	P	R
F	0.739	0.673	0.8	0.752
B	0.414	0.845	0.453	0.676
R	0.704	0.835	0.696	0.879
C	0.25	0.041	0.364	0.054
I	0.6	0.429	0.5	0.571

In Table 2, F represents forward, B bidirection, R reverse, C contradiction and I independence. Run-2 achieves a better

performance than run-1, according to the results shown in the table. For forward relation, an increasing 6.1% performance of precision and 7.9% of recall exists; for bidirection, the increasing performance of precision is 3.9%; for reverse, the increasing performance of recall is 4.4%; for contradiction, the increasing performance is 11.4% for precision and 1.3% for recall, for independence, the increasing performance of recall is 14.2%. However, we also see that some performances drop, such as the recall of bidirection relation. Actually, run-2 comes into a sharp performance drop of bidirection relation in comparison with run-1. Alternately, the recall of forward and independence greatly increase. It indicates that many pairs are correctly judged as bidirection, while some real bidirection pairs, which are correctly judged, are wrongly judged by using two-stage recognition strategy. Nevertheless, the system still achieves an increasing 2.7% performance of accuracy according to the official results.

We can also see that, the system achieves a low performance on judging contradiction relation. It's because most of the features mainly judge the similarity of the strings, while the contradiction relation can also be viewed as a kind of 'similarity' except for some negative words. Therefore, the pairs of contradiction relation are identified as bidirection ones rather than contradiction ones.

4.3 RITE4QA Subtask

For RITE4QA subtask, we submit three runs: RITE1-WHUTE-CS-RITE4QA-01, RITE1-WHUTE-CS-RITE4QA-02 and RITE1-WHUTE-CS-RITE4QA-03. The method we utilized in this subtask is same as the BC subtask, and the following table shows the results.

Table 3. Official results for RITE4QA subtask

	Run-1		Run-2		Run-3	
	P	R	P	R	P	R
Y	0.204	0.8	0.212	0.662	0.203	0.838
N	0.848	0.263	0.841	0.422	0.854	0.223

From the table we can see that, the recall values of entailment relation in all runs are better than the corresponding values of no entailment relation, while the precision values of no entailment relation in all runs are better than the corresponding values of entailment relation, that is, many pairs are incorrectly judged as entailment, whereas they originally belong to no entailment relation. As a matter of fact, the testing data of RITE4QA is derived from a real Question Answering dataset, and the text retrieved in each pair is similar with the hypothesis to a large extent, while the features in our system also mainly estimate the similarity between the text and the hypothesis. Thus many pairs are incorrectly judged as entailment. For a better performance, more precise features or deep semantic analysis approach should be utilized to judge if the hypothesis is semantically entailed by the text.

4.4 Ablation Test

For estimating the contribution of each resource(or feature) to participants' system performances, ablation tests are suggested by the organizer. For this purpose, we make the experiment by removing one feature for each time in run-3 for BC subtask and run-1 for MC subask. Table 4 shows the results of the ablation test.

Table 4. Results of ablation test

System description	Run-3 for BC subtask	Run-1 for MC subtask
With all the features	0.7371	0.5823
Without Word Overlap	0.7224	0.5774
Without Bigram Overlap	0.7076	0.5749
Without Trigram Overlap	0.7273	0.5799
Without Cosine similarity	0.7248	0.5823
Without Euclidean Distance	0.7273	0.5799
Without Jaro Distance	0.7174	0.5872
Without JaroWinkler Distance	0.7273	0.5799
Without LCS Similarity	0.7224	0.5528
Without Character Levenshtein Distance	0.7322	0.5848
Without Word Levenshtein Distance	0.7346	0.5872
Without Length Ratio	0.7125	0.5774
Without Manhattan Distance	0.7199	0.5356
Without Jaccard Coefficient	0.7371	0.5479
Without Chebyshev Distance	0.7248	0.5848
Without Dice Coefficient	0.7199	0.5823
Without Matching Coefficient	0.7346	0.5553
Without Maximal Substring Similarity	0.7224	0.5872

Without Minimal Substring Similarity	0.7273	0.5823
Without Named Entity Coverage	0.6830	0.5767
Without Numeral Coverage	0.7101	0.5725
Without Unlabeled Sub Tree Overlap	0.7273	0.5823
Without Partial Sub Tree Overlap	0.7322	0.5872
Without labeled Sub Tree Overlap	0.7297	0.5799
Without Predicate Argument Overlap	0.7322	0.5823

From table 4 we can see that, Bigram Overlap, Length Ratio, Named Entity Coverage and Numeral Coverage contribute to the performance greatly than others in BC subtask, while Manhattan Distance and Jaccard Coefficient contribute to the performance greatly than others in MC subtask. On the contrary, when removing Jaro Distance, Word Levenshtein Distance, Maximal Substring Similarity and Partial Sub Tree Overlap, the system performance increase slightly. Nevertheless, these features still contribute to an increasing performance in BC subtask so we can still utilize them for judging entailment and no entailment. Furthermore, a more investigation of them should be proceeded if they impact the performance in other entailment dataset.

5. Conclusion

In this paper, we describe our system for RITE subtask at NTCIR-9. We build a textual entailment recognition framework and implement a system that employs string, syntactic, semantic and some specific features for judging entailment relations. To improve the system's performance, a two-stage recognition strategy is utilized, which first judge entailment or no entailment, and then contradiction or independence of the pairs in turn. Official results show that our system achieves a medium performance of all participating system.

We also find that, recognizing contradiction and deep semantic entailment is a direction for improving our system since the performance of the MC subtask is much lower than that of the BC subtask. On the other hand, some features such as named entities contribute to a better performance for recognizing textual entailment; hence another improvement is to apply more available resources for the system.

6. ACKNOWLEDGMENTS

This work is supported by Natural Science Foundation of China(Grant Nos. 61070082, 90820005, 61070243).

7. REFERENCES

[1] Agichtein, E., Askew, W. and Liu, Y. Combining Lexical, Syntactic, and Semantic Evidence For Textual Entailment Classification. In proceedings of the Fourth PASCAL Challenges Workshop on Recognizing Textual Entailment. Gaithersburg, Maryland, USA, 2008.

[2] Androutsopoulos, I. and Malakasiotis, P. 2010. A Survey of Paraphrasing and Textual Entailment Methods. Journal of Artificial Intelligence Research 38: 135-187.

[3] Bar-Haim, R., Berant, J., Dagan, I., Greental, I., Mirkin, s., Shnarch, E. and Szpektor, I. Efficient Semantic Deduction and Approximate Matching over Compact Parse Forests. In proceedings of the Fourth PASCAL Challenges Workshop on Recognizing Textual Entailment. Gaithersburg, Maryland, USA, 2008.

[4] Castillo, J. J. and Alemany, L. A. i. An approach using Named Entities for Recognizing Textual Entailment. In proceedings of the Fourth PASCAL Challenges Workshop on Recognizing Textual Entailment. Gaithersburg, Maryland, USA, 2008.

[5] Galanis, D. and Malakasiotis, P. AUEB at TAC 2008. In proceedings of the Fourth PASCAL Challenges Workshop on Recognizing Textual Entailment. Gaithersburg, Maryland, USA, 2008.

[6] Harabagiu, S. and Hickl, A. Methods for Using Textual Entailment in Open-Domain Question Answering. In proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the ACL. Sydney, Australia, 2006.

[7] Jaro, M. 1995. Probabilistic Linkage of Large Public Health Data File. Statistics in Medicine 14: 491-498.

[8] Lloret, E., Ferrández, Ó., Muñoz, R. and Palomar, M. A Text Summarization Approach under the Influence of Textual Entailment. In proceedings of NLPCS2008. Barcelona, Spain, 2008.

[9] Pado, S., Galley, M., Jurafsky, D. and Manning, C. Robust Machine Translation Evaluation with Entailment Features. In Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP. Singapore, 2009.

[10] Ren, H., Ji, D. and Wan, J. WHU at TAC 2009: A Tri-categorization Approach to Textual Entailment Recognition. In proceedings of the Fifth PASCAL Challenges Workshop on Recognizing Textual Entailment. Gaithersburg, Maryland, USA, 2009.

[11] Ren, H., Ji, D., Wan, J. and Zhang, M. Parsing Syntactic and Semantic Dependencies for Multiple Languages with A Pipeline Approach. In Proceedings of the 13th Conference on Computational Natural Language Learning. Boulder, Colorado, USA, 2009.

[12] Wang, R. and Neumann, G. A Divide-and-Conquer Strategy for Recognizing Textual Entailment. In proceedings of the Fourth PASCAL Challenges Workshop on Recognizing Textual Entailment. Gaithersburg, Maryland, USA, 2008.

[13] Winkler, W. String Comparator Metrics and Enhanced Decision Rules in the Fellegi-Sunter Model of Record Linkage. In proceedings of the Section on Survey Research Methods, 1990.